Module - 3

Part - II

A framework for Distributed Data Analysis for IoT

A Framework for Distributed data analysis for IoT:

- Anomaly Detection
- Problem Statement And Definitions
- Hyper Ellipsoidal Anomaly Detection
- Distributed Anomaly Detection
- Clustering Ellipsoids
- Incremental Local Modeling

A Framework for Distributed Data Analysis for IoT:

- A framework for data analysis as part of a monitoring system where the data are distributed in the network and edge devices have limited capacity in terms of memory and Computational power.
- The main idea of this framework is to support efficient local processing and summarization of the data at the nodes, followed by global processing of the local summaries.
- We introduce this framework within the context of distributed anomaly detection but it can be easily extended to a wider range of tasks.

A Framework for Distributed Data Analysis for IoT:

- The proposed framework covers Three aspects:
  - Local Data Modeling
  - Communication
  - Global Data Modeling.

- The Local Data Modeling → requires an efficient algorithm with low computational cost. The algorithm calculates local summaries of the data and identifies anomalies in the local data.
- Communication: →Only the summaries of the data are communicated to a central location which we refer to as sink. Limiting the communication to only data summaries alleviates the overhead of communicating all the data over the network.
- Global Data Modeling Component: → It is located in a central location, is responsible for finding global summaries and anomalies.

ANOMALY DETECTION

- Anomaly detection is an important unsupervised data processing task which enables us to detect abnormal behavior without having a priori knowledge of possible abnormalities.
- An Anomaly can be defined as "a pattern in the data that does not conform to a well-defined notion of normal behavior".
- This definition is very general and is based on how patterns deviate from normal behavior.
- On this basis we can categorize anomalies in the data into three categories:

A Framework for Distributed Data Analysis for IoT:

Anomalies in the Data - Three Categories:

i. **Outliers**—**Short anomalous patterns** that appear in a **nonsystematic way** in the collected data, usually arising due to **noise** or **faults**;
   - Eg:- due to communication errors.

ii. **Events/Change**—These patterns appear with a **systematic** and **sudden change from previously known normal behavior**.
   - The **duration** of these patterns is usually **longer than outliers**.
   - Eg:- In environmental monitoring, extreme weather conditions are examples of events.
   - The **start of an event** is usually called a **change point.**

iii. **Drifts** — **Slow, unidirectional, long-term changes** in data.
   - Eg;- This usually happens due to the onset of a **fault in a sensor**.

ANOMALY DETECTION

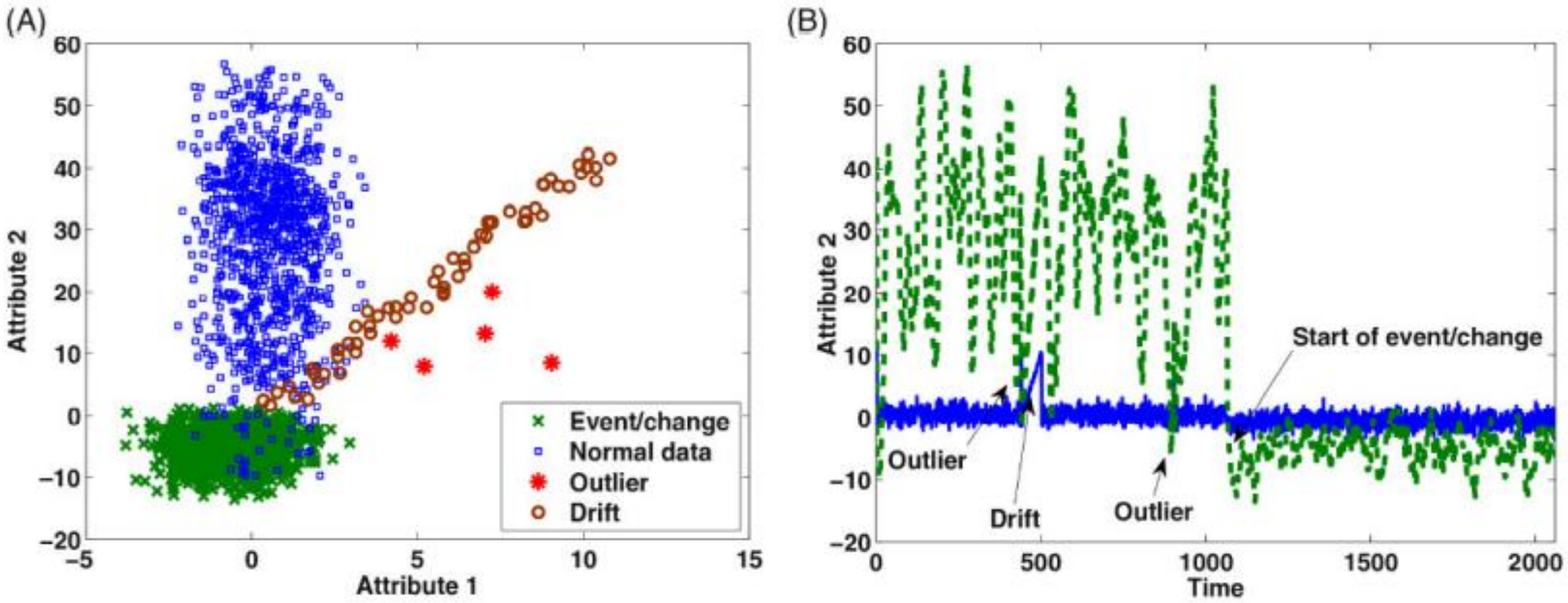


FIGURE 9.2 An Example of Different Categories of Data Anomaly in a Two-Dimensional Dataset

(A) Scatter plot overview. (B) Time series overview.

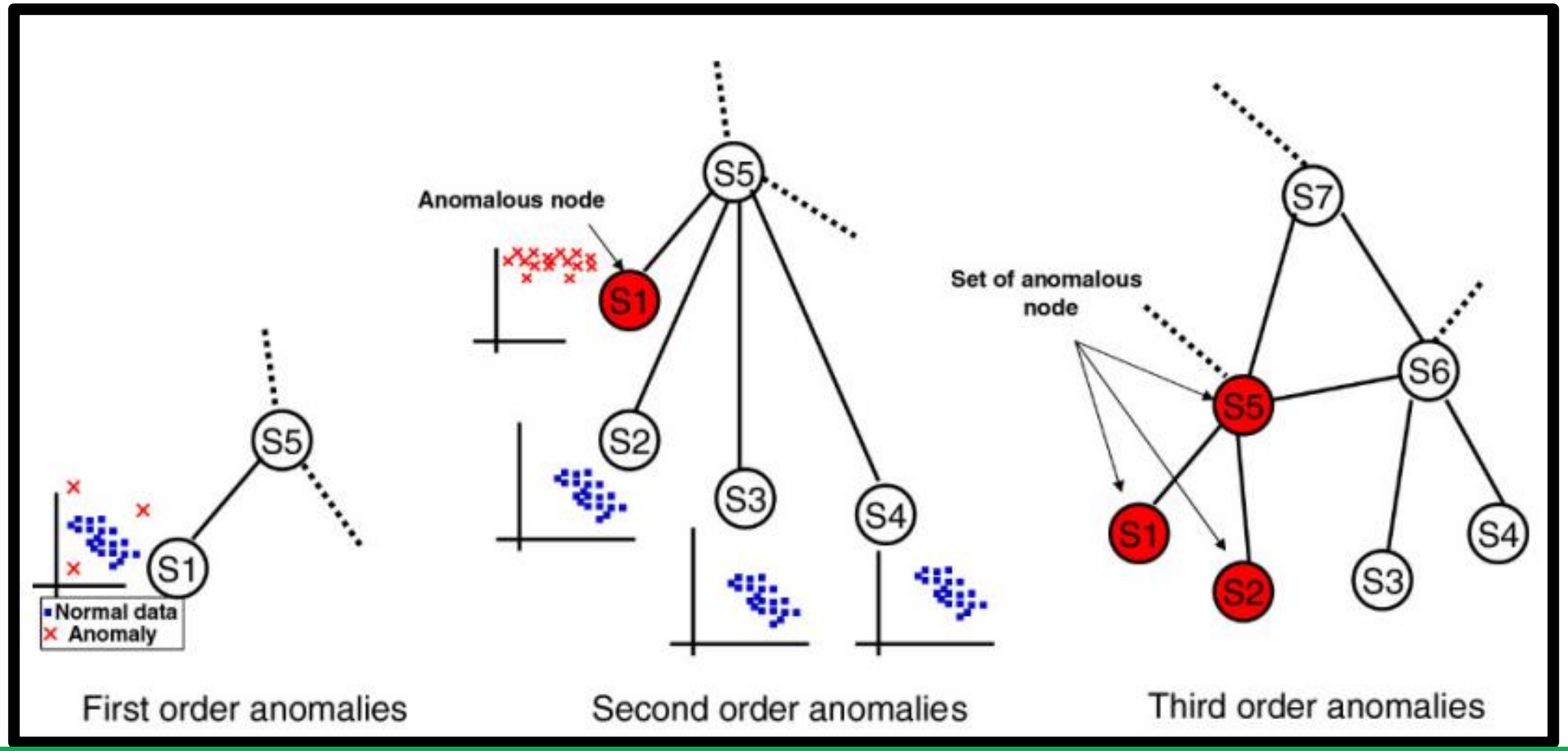
Three Categories Of Anomalies:
Considering The Topology of a Network:

i. **First Order** —Anomalies can occur in an individual node, that is, some observations at a node are anomalous with respect to the rest of the data.

ii. **Second Order** —All of the observed data at a node can be anomalous with respect to the neighboring nodes. In this case, such a node is considered as an anomalous node in the network.

iii. **Third Order** —A cluster of nodes are anomalous with respect to other nodes in the network.

Figure: Types of Anomalies Considering the Topology of a Network:



ANOMALY DETECTION TECHNIQUES

❖ **Many different techniques have been applied for anomaly detection in applications.**

✓ **Some Of The Main Types of Techniques Used In Anomaly Detection:**

i. **The first type of anomaly detection techniques uses Rule-based Methods.**

- **Owing to their simplicity and low computational overhead, these techniques have been successfully implemented in many applications such as Intrusion Detection.**
- **In these approaches, if an observed data sample does not match a predefined set of rules, it is considered as an Anomaly.**

ANOMALY DETECTION TECHNIQUES

ii. The next type of anomaly detection approaches use Dynamic System Modeling to model the normal behavior of the data, and anomalous behavior is then identified by the extent of deviation from the normal model of the data.

- Dynamic Bayesian networks are a common pattern recognition technique used in this area.

ANOMALY DETECTION TECHNIQUES

iii. In Statistical Approaches for anomaly detection, anomalies are considered as those data points that have low likelihood given the expected distribution of the data.

- The main assumption in these approaches is that the distribution of the data is known and the parameters of the distribution need to be estimated using the normal data.
- Density-based and nearest neighborhood approaches identify data points that reside in areas of the input space with low density as anomalous.

iv. One-class Classifiers such as One-class Support Vector Machines have also been used to identify anomalies in a set of data by finding a discriminant between the main body of the data and potential outliers.

v. Clustering Techniques as a Basic knowledge discovery process can also be used for Anomaly Detection on unlabeled data.

ANOMALY DETECTION TECHNIQUES

vi. **Traditional Techniques For Anomaly Detection** the **data** are assumed to be in **one location.**

- However, collecting all the data from the nodes in one location is **not always possible.**
- If we expect to **run anomaly detection locally**, the **processing unit** at each node may not be powerful enough to support sophisticated anomaly detection algorithms.
- Therefore, there is a **need for distributed** and **efficient anomaly detection algorithms suitable for IoT applications**.

ANOMALY DETECTION TECHNIQUES

vii. A Distributed Approach To Anomaly Detection exploits the limited computational resources of the nodes for anomaly detection.

- Nodes can perform the detection locally by collaborating with each other.
- This can save a considerable amount of energy at the nodes by avoiding raw data transmission.
- Therefore, in this case the Benefits Of Using Anomaly Detection Techniques Are Twofold:
  1) The ability to detect interesting samples and
  2) The ability to conserve energy resources at the nodes.

ANOMALY DETECTION TECHNIQUES

viii. **Practical Approach For Distributed Anomaly Detection** is proposed by Rajasegarar et al.

- The authors present a **Hyperellipsoidal Model** for **distributed anomaly detection**, where a **single hyperellipsoid** is **used to determine** the **distribution of all measurements in the network.**
- However, if the monitored environment is **nonhomogeneous**, comprising a mixture of **different distributions**, then this distributed anomaly detection approach will result in **low detection accuracy.**
  - Eg:- This situation can arise in environments, for example, where **some sensors are exposed to direct sunlight, while others lie in shadow.**
  - In this situation, the measurements from each sensor node are **drawn from one of the two different underlying distributions**, and the **anomaly detection algorithm** needs to accommodate this type of **normal behavior.**

PROBLEM STATEMENT AND DEFINITIONS

Aim: To apply an anomaly detection algorithm to the data distributed in a network.

- Consider a sensor network topology in which a set of sensors $N = \{N_j: J = 1,\ldots,l\}$ are connected in a hierarchical tree topology to a sink B.
- Over a fixed monitoring period, each node Nj observes a set of nj measurements $X^j_k = = \{ : k = 1,2\ldots n_j\}$ where each measurement $X^j_k \in \Re^d$ is a vector of values observed at the node;
- Eg:- In a monitoring application these values can be temperature, humidity, and light intensity.
- Our goal is to partition the set X of measurements from all nodes $\cup j=1,\ldots,$ Xj into normal measurements NPX and anomalous measurements APX;
- i.e, $X = NPx \cup APx$, $NPx \cap Apx = \emptyset$.

HYPERELLIPSOIDAL ANOMALY DETECTION

- Hyperellipsoids are remarkably flexible and can capture the central tendencies of a wide range of datasets.
- We introduce the formulas for calculating a hyperellipsoidal decision boundary to detect anomalies.
- Let x be an observation of the random vector $X = (X^1, \dots, X^d)^T \sim X(\mu, \Sigma)$ with population mean $\mu=(\mu 1,\dots,\mu d)^T$ and (positive definite) population covariance matrix $\sum = [cov(X)]$.
- For fixed $\mu \in \Re d$, the level set of $Q(x- \mu)= (X-\mu)^T \ \Sigma^{-1} (X-\mu) = || (X-\mu) ||^2_{\Sigma-1}$
- for scalar $t^2 > 0$ is

$$\text{surf}(\Sigma^{-1}, \mu; t) = \{\mathbf{x} \in \Re^d \mid \|\mathbf{x} - \mu\|^2_{\Sigma^{-1}} = t^2\}$$

- where $\sum^{-1}$ is sometimes called the characteristic matrix of Q, and
- $$\|\mathbf{x} - \mu\|_{\Sigma^{-1}} = \sqrt{(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)}$$ → is the statistical (Mahalanobis) distance between x and μ.

HYPERELLIPSOIDAL ANOMALY DETECTION

- Geometrically,
- surf($\sum^{-1}$,μ;t) →surface of the hyperellipsoid (ellipsoid) in d-space induced by $\sum^{-1}$, all of whose points are at a constant Mahalanobis distance (t) from its center μ.
- The parameters of this ellipsoid are μ, ∑, and t.

HYPERELLIPSOIDAL ANOMALY DETECTION

- We can estimate μ and ∑ using the sample mean and sample covariance matrix at each node N_j ,

- $$\mathbf{m}_j = \sum_{k=1}^{n_j} \mathbf{x}_k^j / n_j = (m_1, \ldots, m_d)^{\mathrm{T}}$$

- $$S_j = \sum_{k=1}^{n_j} (\mathbf{x}_k^j - \mathbf{m}_j)(\mathbf{x}_k^j - \mathbf{m}_j)^{\mathrm{T}} / (n_j - 1)$$

HYPERELLIPSOIDAL ANOMALY DETECTION

- Now, we can define **Normal** and **Anomalous Measurement** relative to the **ellipsoidal parameters** (S_j^{-1} ,m_j) as :

$$\mathrm{NP}_{X_j} \equiv \mathrm{NP}_{X_j}(S_j^{-1}, \mathbf{m}_j ; t) = \{\mathbf{x}_{j,k} \in \mathrm{X}_j \mid \left\| \mathbf{x}_\mathbf{k} - \mathbf{m}_j \right\|_{S_j^{-1}}^2 \leq t^2\} \sim (\text{normal points in } X_j)$$

$$\mathrm{AP}_{X_j} \equiv \mathrm{NP}_{X_j}(S_j^{-1}, \mathbf{m}_j ; t) = \{\mathbf{x}_{j,k} \in X_j \mid \left\| \mathbf{x}_\mathbf{k} - \mathbf{m}_j \right\|_{S_j^{-1}}^2 > t^2\} \sim (\text{anomalous points in } X_j)$$

Distributed Anomaly Detection

- In this section, we generalize the distributed anomaly detection approach proposed by S Rajasegarar, to the case of learning a multimodal global model of normal behavior in the network.
- We first model the normal data of each node using the ellipsoidal boundary described.
- We then communicate the parameters of the hyperellipsoid from each node to the sink, where we cluster these 'l' hyperellipsoids to c clusters that reflect the global distribution of measurements in the network.
- The c merged hyperellipsoids corresponding to these clusters are then reported back to the nodes where anomaly detection can be performed.
- The final step is based on the idea that, at some point in time all the nodes will observe all the modes in the environment.
- In some applications, the network may have multiple subsections which are expected to have different characteristics.
- In these cases, this algorithm should be run independently within each subsection.
- The steps of the algorithm are shown in the following Box.

ALGORITHM 9.1 DISTRIBUTED ANOMALY DETECTION BY CLUSTERING ELLIPSOIDS

Step 1—At each node $N_j \in N$

- Calculate the local ellipsoid e_j from X_j
- Transmit parameters of e_j to the sink B

Step 2—At base station B

- Receive ellipsoid parameters e_j from each node
- Calculate similarity $s(e_j, e_i)$ between all pairs of ellipsoids
- Estimate the number of clusters c among the ellipsoids $E = \{e_j | j = 1,\ldots,l\}$
- Cluster ellipsoids E into c merged ellipsoids $E' = \{e'_r | r = 1,\ldots,c\}$
- Transmit parameters of merged ellipsoids E' to each sensor $N_j \in N$

Step 3—At each sensor

- Use merged ellipsoids E' to detect global anomalies by marking any observation that falls outside all the merged ellipsoids.

CLUSTERING ELLIPSOIDS

- It is a clustering approach to group similar ellipsoids together instead of simply merging all the ellipsoids into one ellipsoid.
- The main purpose of clustering ellipsoids is to remove redundancy between the ellipsoids reported by the nodes that have the same underlying distributions.

FIGURE 9.4:

Global Ellipsoidal Boundaries for the Cases of a Single Global Ellipsoid (c=1) and Multiple Global Ellipsoids (c = 2) After Clustering.

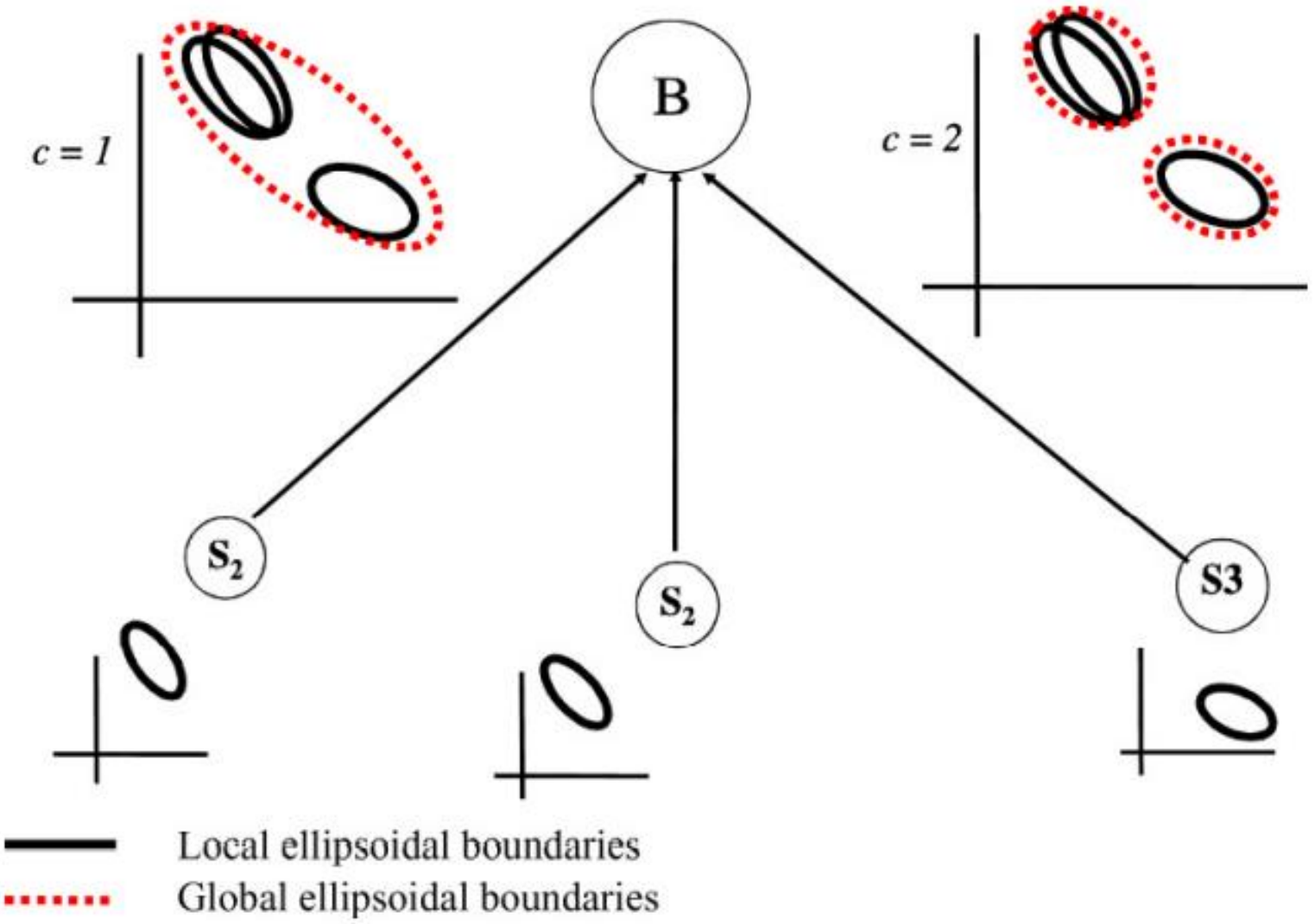


FIGURE 9.5 : Example of Global Measurements in a Nonhomogeneous Environment (Normal Measurements are Represented by Squares, Whereas Anomalies Introduced Into the Data are Represented by Crosses)

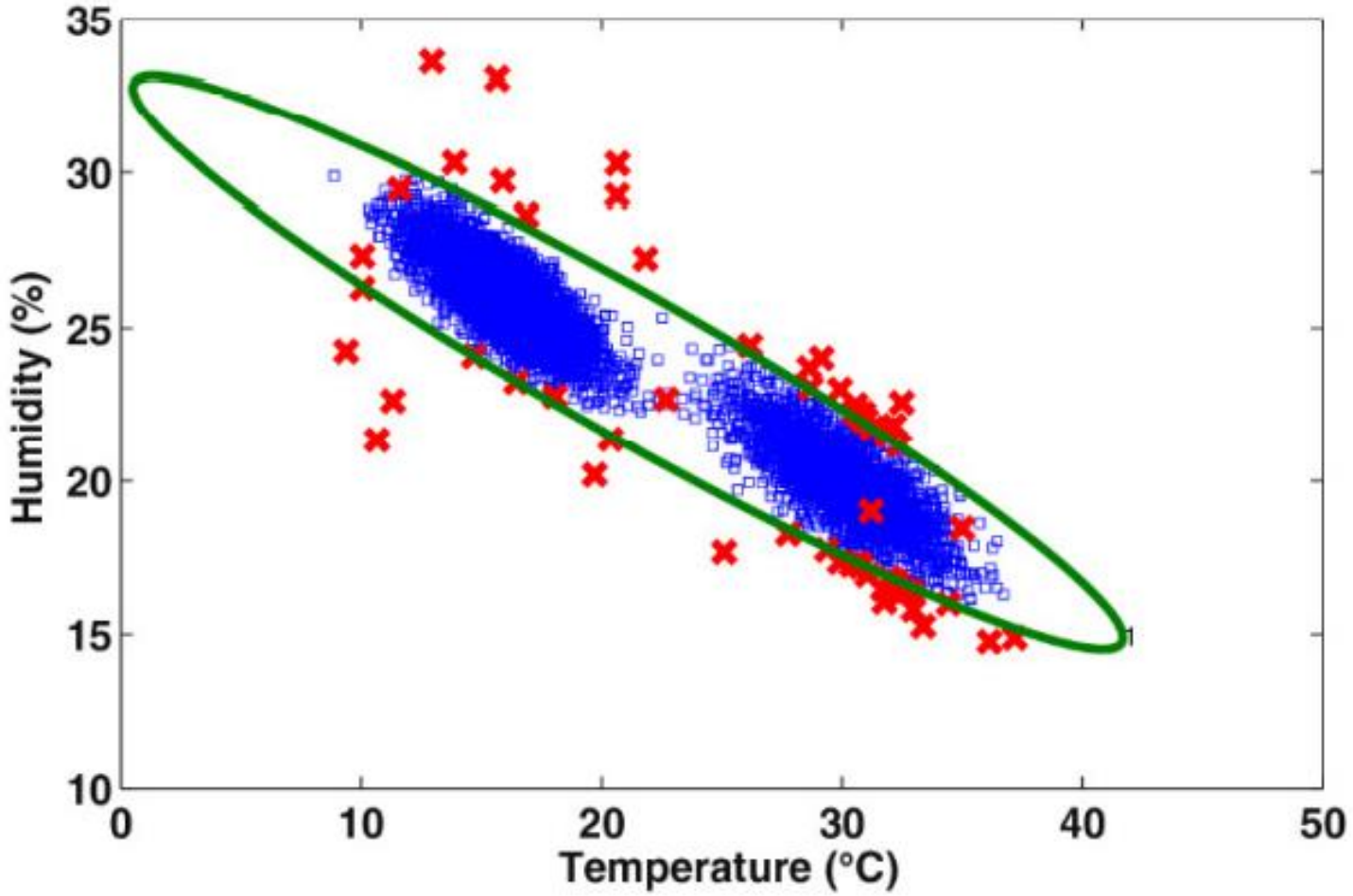


Figure shows measurements from two groups that have distinct and separate distributions.

CLUSTERING ELLIPSOIDS

- Trying to fit a single ellipse (c = 1) to these data results in underfitting, whereas two global ellipses (c = 2) can provide an accurate fit for normal measurements.
- To determine global anomalies in nonhomogeneous environments, each node Nj sends the parameters – (S_j^{-1} , m_j, n_j) of its local ellipsoid e_j to the sink B.
- As shown in Box 9.1, the sink gathers these parameters from each node, and clusters the ellipsoids.
- If needed, the sink can report the parameters of the clustered ellipsoids back to the nodes to provide them with a set of global decision boundaries.
- We now describe the main steps in more detail.

Steps:

1. Calculating similarities between ellipsoids.
2. Estimating the number of clusters.
3. Clustering ellipsoids.

1. Calculating similarities between ellipsoids.

- To compare and cluster ellipsoids, we first need a similarity or distance measure between a pair of ellipsoids.
- We use a simple similarity measure based on the distance between the centers of the ellipsoids.
- Let m1 and m2 be the centers of ellipsoids e1 and e2 , respectively.
- The similarity function s(e1, e2) is as follows:
-

$$s(e_1, e_2) = e^{-\|m_1 - m_2\|}$$

2. Estimating the Number of Clusters

- Before clustering the ellipsoids $E = \{e_j | j = 1,..,l\}$, we need to estimate the number of clusters c.
- A nonsymmetric n × n matrix consisting of c blocks (clusters) has c large eigenvalues of order c while the other characteristic values remain of order n as n tends to infinity.
- A similarity matrix that can be perfectly divided into c clusters would have a c block diagonal shape.
- Using this theorem as a preclustering assessment method to help when choosing the best number of clusters, by looking for a "big jump" in a plot of the square roots of the ordered eigenvalues (called a PRE plot) of a similarity matrix S (square roots just improve the visual interpretation of where the big jump occurs).
- *Note that the theory underlying this strategy is not tied to any clustering algorithm.*

CLUSTERING ELLIPSOIDS

- A PRE plot diagram of a similarity matrix can be characterized by an initially high negative gradient followed by a horizontal gradient.
- The point on the x-axis where these two gradients intersect yields the number of clusters c.
- To detect this change in the PRE plot, the gradient between each pair of consecutive points on the plot is calculated, and a large difference between consecutive gradient values is used to choose the value of c where there is a change in the order of magnitude of the eigenvalues.
- Note that the calculation of eigenvalues is performed at the sink, and not at the nodes.
- The sink is not considered to be constrained in computational power

3. Clustering Ellipsoids

- Having defined a similarity measure between ellipsoids, the given set of ellipsoids E can be clustered using bottom-up hierarchical clustering with single linkage.
- For this algorithm, the parameter for the number of clusters 'c' in the hierarchical clustering is based on the value derived from the PRE plot as described.
- The clustering partitions E into c sets of ellipsoids, where each set of ellipsoids in a partition needs to be merged into a single ellipsoid, that is, $E \rightarrow E' = \{e'_k \mid k = 1,\ldots,c\}$.

CLUSTERING ELLIPSOIDS:

- Ellipsoids can be merged in a pair wise manner as follows. Let – (S_i ,n_i , m_i) , and (S_j ,n_j , m_j) be the parameters of ellipsoids e_i and e_j , then the parameters (S^{-1} ,m, n) , of the ellipsoid é' derived from merging ei and ej are:

$$n = n_i + n_j$$

$$m = \frac{n_i}{n} m_i + \frac{n_j}{n} m_j$$

$$S = \frac{n_i - 1}{n - 1} S_i + \frac{n_j - 1}{n - 1} S_j + \frac{n_i n_j}{n(n-1)} \left[(m_i - m_j)(m_i - m_j)^{\mathrm{T}} \right]$$

Eq. of 'S' is given to merge two sample covariance matrices and not the inverse of the sample covariance matrices, which are transmitted by the nodes. The sink should perform the necessary inverse operations to calculate S^{-1}. After merging, the merged ellipsoids E' can be transmitted back to the sensor nodes to detect global anomalies.

EFFICIENT INCREMENTAL LOCAL MODELING.

- One of the Drawbacks of the proposed approach is that its local modeling is performed in batch mode.
- Each node has to buffer a window of measurements, then calculates the local ellipsoidal boundary, and sends it to the sink.
- The anomaly detection can be done according to both local and global ellipsoids at the node.
- However, selecting an appropriate window is known to be a difficult task, as a small window may result in an inaccurate estimate of the local ellipsoid, and the data in a larger window size might not come from a unimodal distribution, which contradicts the assumption of the model.
- Another consideration about the batch calculation of the parameters of the local model is the limited computational and memory capacity of each node.
- The computational complexity of calculating the parameters of the ellipsoidal model (especially in low dimensions) is considered to be low, but when done in batch mode can restrict the function of the node for an extended time.
- Many current nodes in IoT networks have very limited memory, which limit the practicality of the batch approach for the computation of the local model.

EFFICIENT INCREMENTAL LOCAL MODELING.

- In this section, we describe an **incremental approximation of the batch local ellipsoids** that can address the aforementioned shortcomings of the local batch calculation of the ellipsoidal model. This approach:
  1. Does not require a window size;
  2. Enables the node to have an **updated local ellipsoidal boundary** at any point in time to potentially remove anomalies before incorporating them into the model;
  3. Breaks down the batch calculation of the ellipsoids to smaller updates after each sample using an efficient incremental update formula.

**** (The method incrementally updates its clusters as new data come in online).*

INCREMENTAL UPDATES

- The parameters of the local ellipsoid at each node N_j at time n – ($S_{j,n}^{-1}$,$m_{j,n}$) can be estimated incrementally (sometimes called “recursively”).
- Incremental updates for the mean and covariance matrix at Eqs. (9.2) and (9.3) are (for clarity of exposition we dispense with the index j).

$$\hat{\mathbf{m}}_n = \hat{\mathbf{m}}_{n-1} + \frac{(\mathbf{x}_n - \hat{\mathbf{m}}_{n-1})}{n}$$

$$\hat{S}_n = \left(\frac{n-1}{n}\right)\hat{S}_{n-1} + \frac{(\mathbf{x}_n - \hat{\mathbf{m}}_{n-1})(\mathbf{x}_n - \hat{\mathbf{m}}_{n-1})^{\mathrm{T}}}{n}$$